Modeling hybrid virtual objects using scalable multi-modal components and rigid transformations

Zongcheng Li[1, 2], Franca Giannini[2], Jean-Philippe Pernot[1], Philippe Véron[1], Bianca Falcidieno[2]

1Arts et Métiers ParisTech
LSIS Laboratory UMR CNRS 7296
Aix-en-Provence, France
{firstname.lastname}@ensam.eu

2IMATI-CNR
Genova, Italy
{firstname.lastname}@ge.imati.cnr.it

Rapporto tecnico IMATI n.4/2014
Genova, 2.9.2014

INTRODUCTION

The specification of digital objects, assemblies and scenes strongly relies on the use of constraints describing how shapes have to be combined all together. Constraints can be applied to the low level geometric entities or the parameters of the geometry, as in [Hsu97] and [And95]. They are also used to set position relations between different geometries considered as rigid bodies as in [Mar08] and [Kra91]. In both cases, they act on the same type of geometry either 2 or 3D. In this paper, we extend the classical paradigm to a more general concept so as to be able to apply them simultaneously on multimodal data, i.e. on different digital representations (e.g. 2D raster image, 3D point cloud, 3D mesh) obtained through different approaches (e.g. drawing, picture, 3D modeling, 3D scanning). Actually, the accessibility to a great amount of digital graphical data offers a rich space of inspiration resources for creative users and designers. To exploit at the best such resources for browsing and communicating new ideas, tools need to be developed enabling to easily combine data independently of their origin and type and to fast mock up new solutions. New ideas frequently originate from rearranging some existing elements. It could be a combination of different concepts or a merging of different domains, such as the objects in Fig. 1. Interesting elements can be seen and be available in different formats, e.g. they may come from a picture or a 3D model, that the designer has seen before. Thus, if we fix the modality of data to design an object, it will limit the re-use of available data to modeling an object that directly express the initial ideas of the user.

Fig. 1 Example of unconventional chairs

Despite the multimodality of representation, all of them provide the geometric information of the shape, i.e. the spatial extent of the object. Based on the geometric information, a lot of research works have been done to help understand the structure of the object. For example, the Reeb Graph (such as [Har09], [Bia08]), the skeleton (such as [Abo91], [Lak07] and [Tao07]), have been widely exploited for classifying different objects or for segmenting an object in meaningful elements [Ber09, LGD10]. Being interested in creating new objects from parts of existing ones, their structural descriptors can be directly exploited for selecting and positions the part of interest. Applying the constraints on the structural level (e.g. skeleton) of a shape might be more useful and more meaningful from the user's point of view. Additionally, once chosen a specific method (e.g. reeb graph vs skeleton), the structure of a shape is independent from the geometric representation.

Based on the above considerations, this paper proposes new approach to the modeling of objects from multimodal data. The final objects are defined from a set of multimodal data constrained together on both geometric and structural levels. The final descriptions are obtained while solving an optimization problem using a specific energy function to be minimized. Section 2, provides an overview of the proposed approach. Section 3 introduced all the key entities lying on multimodal data on which constraints can be applied. The different types and levels of constraints are presented in section 4. The adopted *constraint satisfaction method* is introduced in section 5. Section 6 shows some results as well as the implementation. The conclusion and the perspectives are discussed in section 7.

OVER VIEW OF THE PROPOSED APPROACH

In this paper, we propose an approach for constraining scalable components represented by multimodal data. To explain this approach, three questions needed to be answered at first:

- What to constrain?
- Which constraints could be applied?
- How to evaluate the constraints?

Given a subpart (component) of our multimodal data, what we need to constrain is a tuple (*P*, *R*, *S)*. *P* is the **position** of the component in a global space, defined as the position of the origin of the local reference frame of the component in global space. *R* is **orientation** of the component, defined as the orientation of the local reference frame in a global space. *S* is the **scale factor** along the three axis of the local reference frame. Then constraining two components can be translated into the problem of constraining each local reference frame in a global space.

Since our aim is to support non-experts, constraints should be referred to more meaningful component elements than the local reference frame. We call these elements: **Key Entities**. They correspond to significant elements derived from the geometry of a component. They are evaluated in the local reference frame of the component. Thus, the constraints establish the link between different local features on the local reference frames. After having specified the constraints between different components, a *constraint satisfaction problem (CSP)* has to be solved to determine the final object.

KEY ENTITY ON MULTIMODAL DATA

Two different types of key entity are proposed: **Geometric Key Entity ($\boldsymbol{E_g}$)** and a **Parametric key Entity ($\boldsymbol{E_p}$)**. The geometric key entity corresponds to a position in the space, while a parametric key entity refers to specific elements on the geometric or the structural representation of the component, possibly computed from specific rules and parameters.

A Component of a multimodal data can be formally described as follows:

$\boldsymbol{Com} \coloneqq \boldsymbol{Rf} \times \boldsymbol{Geo} \times \boldsymbol{Str}, where$

$\boldsymbol{Rf} \coloneqq \boldsymbol{P} \times \boldsymbol{R} \times \boldsymbol{S}$

Rf is the reference frame of the component composed by the origin (***P***) – a 3D Point, rotation (***R***) and scale (***S***).

$\boldsymbol{P} \coloneqq (x, y, z), where\ x, y, z \in \boldsymbol{R}$

$\boldsymbol{R} \coloneqq (\alpha, \beta, \gamma), where\ \alpha, \beta, \gamma \in [0, \pi] \subseteq \mathbf{R}$

$\boldsymbol{S} \coloneqq (a, b, c), where\ a, b, c \in \mathbf{R}$

Geo is the geometric representation of this component, which could be different according to different types of data.

Here we consider the geometry of a 3D component as a mesh being a format largely available in the web, always achievable (either from CAD data or point clouds) and on which tools for structure extraction and segmentation are being developed. Thus, for a 3D component, the geometric information could be described as below:

$\boldsymbol{Geo_{3D}} \coloneqq (Vertices, Triangles)$

$Vertices = \{vertex_i\}_{t \in [0,m]},$

$where\ vertex_i \in \boldsymbol{P}, i, m \in \mathbf{N}$

$Triangles = \left\{triangle_j\right\}_{j \in [0,n]},$

$triangle_j = (vertex_{j0}, vertex_{j1}, vertex_{j2})$

$where\ \ vertex_{j0}, vertex_{j1}, vertex_{j2} \in Vertices,$

$j0 \neq j1 \neq j2\ and\ j, n \in \mathbf{N}$

For a 2D Component, for constrain purposes, we are considering the contours of the shape, which can be obtained from any 2D raster or vector data [Pap11, PGY99, Kir13, Mor12]. So we could structure the 2D geometric information as below:

$\boldsymbol{Geo_{2D}} \coloneqq \{point_i\}_{i \in [0,h]},$

$where\ point_i = (x_i, y_i, 0) \in \boldsymbol{P}, i, h \in \mathbf{N}, \qquad x, y \in \mathbf{R}$

Where *points* is the list of points on the contour with a clockwise order.

Str is the structural representation of the component such as the reeb graph, the medial axis, the skeleton, etc. They are normally represented as graphs, thus the structural information can be represented as below:

$\boldsymbol{Str} \coloneqq (Nd, Ar)$

$Nd = \{node_i\}_{i \in [0,s]}, where\ i, s \in \mathbf{N},$

$node_i \in \boldsymbol{P}$

$Ar = \{arc_j\}_{j \in [0,t]}, arc_j = (node_{j0}, node_{j1})$

$where\ node_{j0}, node_{j1} \in Nd, j0 \neq j1, j, t \in \mathbf{N}$

To be accessible for the user, each node is associated to a 3D point in the component local reference frame and corresponds to specific shape characteristic points, e.g. the center of the section of the change of the considered function for the reeb graph computation [Bia08].

We have identified as meaningful 18 types of key entities: 5 are geometric key entities (E_{gxy}) and 13 types of parametric key entities (E_{pxy}). They are summarized in table 1 where *x* indicates the type of entity the constraint refers to: "Rf" for the reference frame, "p" for a point, "l" for a line, "f" for planar face, "g" for a whole geometry and "a" for a list of parameters. The index *y* is is used to differentiate same category of constraints acting on the same type of entities. In the following the various key entities are described.

E_{gRf}	The local reference fame
E_{gp}	A point
E_{gl}	A line
E_{gf}	A plane
E_{gg}	The whole geometry
E_{pp1}	A point on an 2D contour
E_{pp2}	A interpolated point on a 2D contour
E_{pp3}	A point inside of the 2D contour
E_{pp4}	The center of a 2D contour
E_{pp5}	A node on the structure
E_{pp6}	A interpolated node the structure
E_{pp7}	A vertex of a mesh
E_{pp8}	A point on the mesh surface
E_{pp9}	The center of a 3D geometry
E_{pl1}	An edge on the contour
E_{pl2}	An arc of a structure
E_{pl3}	The middle axis of the MBB
E_{pa}	An array of other key entities

Table 1. Different types of Key Entity

$\boldsymbol{E_{gRf}} \coloneqq rf$, that is the local reference frame rf.

$\boldsymbol{E_{gp}} \coloneqq (rf, P), where\ P \in \boldsymbol{P}$, that is a point in the local reference frame.

$\boldsymbol{E_{gl}} \coloneqq (rf, Pl, \overrightarrow{nl}), where\ Pl \in \boldsymbol{P}\ and\ \overrightarrow{nl}\ is\ versor$

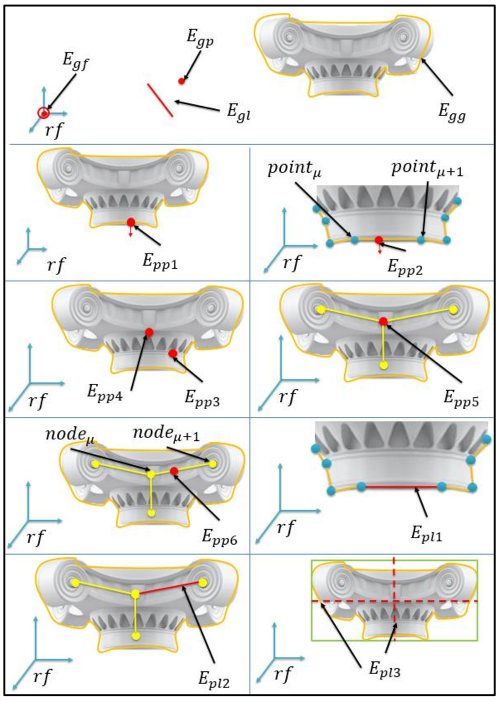

Fig. 2 Examples of 2D Key Entity

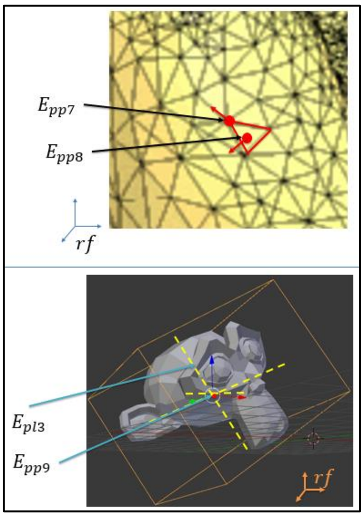

Fig.3 Examples of 3 3D Key Entity

pl is one point on the line with direction $\overrightarrow{nl}$.

$\boldsymbol{E_{gf}} \coloneqq \left(rf, Pf, \overrightarrow{nf}\right),$

$where\ Pf \in \boldsymbol{P}\ and\ \overrightarrow{nl}\ is\ a\ versor$

Pf is one point on the plane with $\overrightarrow{nf}$ normal.

$\boldsymbol{E_{gg}} \coloneqq (rf, geo)$

Similarly for the parametric key entities of a 2D component we have:

$\boldsymbol{E_{pp1}} \coloneqq (rf, geo_{2D}, \mu) \to \left(P_{\mu}, \vec{n}_{\mu}\right)$

P_{μ} is the position of the μth point on the contour

$\vec{n}_{\mu 1} = \left\| \frac{(P_{\mu}-P_{\mu+1})+(P_{\mu}-P_{\mu-1})}{2} \right\|$ is the normal of the μth point on the contour which is very useful because for the 2D component, the contour represents the surface of this component. And the normal of the contour point is actually the normal of the surface of a shape in 2D situation.

$\boldsymbol{E_{pp2}} \coloneqq (rf, geo_{2D}, \mu, f) \to (P, \vec{n})$

$$P = P_{\mu} + f \cdot \left(P_{\mu+1} - P_{\mu}\right)$$

$$\vec{n} = (1-f) \cdot \vec{n}_{\mu} + f \cdot \vec{n}_{\mu+1}$$

$\boldsymbol{E_{pp3}} \coloneqq (rf, geo_{2D}, x, y)$

$$\to P = \frac{1}{\overline{N}} \sum_{i=0}^{i=\overline{N}-1} P_i + (x, y, 0)$$

$$\boldsymbol{E_{pp4}} \coloneqq (rf, geo_{2D}) \to P = \frac{1}{h} \sum_{i=0}^{i=h-1} P_i$$

$\boldsymbol{E_{pp5}} \coloneqq (rf, str, \mu) \to P = nd_{\mu}$

$\boldsymbol{E_{pp6}} \coloneqq (rf, str, \mu, f)$

$$\to P = nd_{\mu} + f \cdot (nd_{\mu+1} - nd_{\mu})$$

$\boldsymbol{E_{pl1}} \coloneqq (rf, geo_{2D}, \mu) \rightarrow (P, \vec{n}),$

$P = 0.5 \cdot (P_{\mu+1} + P_\mu),$

$\vec{n} = \frac{P_{\mu+1} - P_\mu}{\|P_{\mu+1} - P_\mu\|},$

$where\ P_\mu, P_{\mu+1} \in geo_{2D}, \mu \in [0, h-1] \subseteq \mathbf{N},$

$\boldsymbol{E_{pl2}} \coloneqq (rf, str, \mu) \rightarrow (P, \vec{n}_\mu)$

$$P = 0.5 \cdot (nd_{\mu 1} + nd_{\mu 0});$$

$$\vec{n}_\mu = \frac{nd_{\mu 1} - nd_{\mu 0}}{\|nd_{\mu 1} - nd_{\mu 0}\|},$$

$where\ \ \mu$ is the index of the arc in the structure and $nd_{\mu 0}, nd_{\mu 1}$ are the index of the node at the two edges of this arc.

$\boldsymbol{E_{pl3}} \coloneqq (rf, geo, \mu) \rightarrow (P, \vec{n}_\mu),$

is one of middle axis of the minimum

bounding box (MBB) of the component, $\mu \in [0,2] \in \mathbf{N}$, P is the center of the MBB.

$\boldsymbol{E_{pa}} \coloneqq \{E_i\}_{i \in [0,n]}, where\ i, n \in \mathbf{N}$, E_i is one of the other key entity described above.

For a 3D component, for the parametric key entities we consider the correspondent in 3D For sake of space we only list them in table 1 and present graphical examples in figure 3.

Although the key entities apply to geometry of different dimensions and are all calculated in the local reference frame, they are anyhow immersed in the same 3D global coordinate system.

Another important point we can see here is that for one single key entity, sometimes we can get more than one 3D elements. For e.g. the $\boldsymbol{E_{pp8}}$, we have 6 parameters: $rf, geo_{3D}, \mu, d0, d1$ and $d1$. Form those six parameters we can obtain two 3D elements. The first 3D element we can get is the position of this vertex on the mesh, and the second 3D element is the normal of this vertex. In this sense, key element has its own dimension. $\boldsymbol{E_{pp8}}$ has two dimensions. According to the dimension, we can classifier all kinds of key entities in the table 2. We have one type of 1D key entity, three types of 2D (line, plane, and point-vector) we use 2D1, 2D2, 2D3 to separate the different types, which will be used in the next section to classify the different combination of two key entities.

CONSTRAINTS

Constraints actually are equations built between different key entities. So to define a constraint, we need to define which are the key entities involved, and also what equations to be applied. According to the number of equations we classified constraints into two levels. Low level with only one equation and high level with more than one equation.

1D	A Point		$E_{gp}, E_{pp3}, E_{pp4}, E_{pp5}, E_{pp6}, E_{pp9}$
2D	2D1	A Line	$E_{gl}, E_{pl1}, E_{pl2}, E_{pl3}$
	2D2	A Plane	E_{gf}
	2D3	A Point-Vector	$E_{pp1}, E_{pp2}, E_{pp7}, E_{pp8}$
3D	Reference frame		E_{gRf}
nD	nD1	The whole geometry	E_{pg}
	nD2	A list of Key Entities	E_{pa}

Table 2. Classification of Key Entities

In this paper, we are considering that constraints are built only between two key entities (if there are more than two key entities we can use E_{pl}). So the possible combinations of two key entities for one constraint are indicated in Table 3. We can see there are 28 different numbered situations.

Constraints could be applied to one or more of those situations. In different situations, the equations of the constraint might be different. In table 4. We list all the constraints that we are considering, with the possible situations of combination of key entities (see Table 3).

	1D	2D1	2D2	2D3	3D	nD1	nD2
1D	1						
2D1	2	3					
2D2	4	5	6				
2D3	7	8	9	10			
3D	11	12	13	14	15		
nD1	16	17	18	19	20	21	
nD2	22	23	24	25	26	27	28

Table 3 Combination of key entities for one constraint

Before defining all the constraints, we need to put the key entities in a same space: the global space, we indicate the transformation matrix applied to a key entity I as M_i .

In the following definitions we use E_1, E_2 to present the two key entities in a constraint, the order is from the row to the column as presented in Table 3. For e.g. for Combination 7, E_1 is a key entity of type 2D3 and E_2 is a key entity of type 1D. Then we use E_1^0, E_1^1, E_1^3 to present the first, second and third 3D element of key entity E_1. For e.g. if E_1 is of type 2D3 then, E_1^0 presents the position of this vertex and E_1^1 presents the normal of this vertex. Table 4 contains the list of the considered constraints; *Cl* indicates a low level constraint and *Ch* indicates a high level constraint. In the following description of the constraints, the indices specify the key entity combination considered. :

No.	Name	Acceptable Key entity Combination
Cl1	Distance	1,2,4,7,8,9,10,11,12,13,14,15,16,18,19,20,21
Cl2	Angle	3,5,6,8,9,10
Cl3	Coincidence	1,7,10,11,14,15,16,20
Cl4	Parallel	3,5,6,8,9,10
Cl5	Perpendicular	3,5,6,8,9,10
Cl6	Collinear	2,8,12
Cl7	Coplanar	4,9,13
Ch1	Coaxial	
Ch2	Tangent	5,6,8,9,10,
Ch3	Insert	3
Ch4	Contact	18,19,21
Ch5	Pattern	22,23,25,26

Table 4. Constraints

$Cl1_{1,7,10,11,14,15} \coloneqq \|M_2 \cdot E_2^0 - M_1 \cdot E_1^0\| \ == constant > 0$

$$Cl1_2 \coloneqq \frac{\left\|(M_2 \cdot E_2^0 - M_1 \cdot E_1^0) \times (\left[{M_1}^{-1}\right]^T \cdot E_1^1)\right\|}{\left\|\left[{M_1}^{-1}\right]^T \cdot E_1^1\right\|} == constant > 0$$

$$Cl1_4 \coloneqq \frac{\left\|(M_2 \cdot E_2^0 - M_1 \cdot E_1^0) \cdot (\left[{M_1}^{-1}\right]^T \cdot E_1^1)\right\|}{\left\|\left[{M_1}^{-1}\right]^T \cdot E_1^1\right\|}$$
$$== constant > 0$$

$Cl1_{8,12}(E_1, E_2) \coloneqq Cl1_2(E_2, E_1)$

$Cl1_{9,13}(E_1, E_2) \coloneqq Cl1_4(E_2, E_1)$

$Cl1_{16,18,19,20} \coloneqq \left\|M_2 \cdot E_2^0 - M_1 \cdot E_1^i\right\| == constant > 0$

$where\ \exists\ i\ \in N, for\ \forall p \in N, p \neq i, E_1^i, E_1^p \in E_1,$

$i, p \in [0, n] \subseteq N,$

n is the number of geometrc elements

of the component

$\rightarrow \left\|M_2 \cdot E_2^0 - M_1 \cdot E_1^i\right\| \leq \left\|M_2 \cdot E_2^0 - M_1 \cdot E_1^p\right\|$

$Cl1_{21} \coloneqq \left\|M_2 \cdot E_2^i - M_1 \cdot E_1^j\right\| == constant > 0$

$where\ \exists\ i, j\ \in N, for\ \forall p, q \in N, p \neq i, q \neq j,$

$E_2^i, E_2^p \in E_2, E_1^j, E_1^q \in E_1,$

$, i, p \in [0, n] \subseteq N, j, q \in [0, m] \subseteq N,$

n, m is the number of geometrc elements

of the component 2 and component 1

$\rightarrow \left\|M_2 \cdot E_2^i - M_1 \cdot E_1^j\right\| \leq \left\|M_2 \cdot E_2^p - M_1 \cdot E_1^q\right\|$

$Cl2_{3,6,9,10} \coloneqq$

$$\cos^{-1}\frac{[{M_2}^{-1}]^T \cdot E_2^1 \cdot [{M_1}^{-1}]^T \cdot E_1^1}{\left\|[{M_2}^{-1}]^T \cdot E_2^1\right\| \cdot \left\|[{M_1}^{-1}]^T \cdot E_1^1\right\|}$$

$$== constant \in [0, \pi]$$

$Cl2_{5,8} \coloneqq$

$$\pi - \cos^{-1}\frac{[{M_2}^{-1}]^T \cdot E_2^1 \cdot [{M_1}^{-1}]^T \cdot E_1^1}{\left\|[{M_2}^{-1}]^T \cdot E_2^1\right\| \cdot \left\|[{M_1}^{-1}]^T \cdot E_1^1\right\|}$$

$$== constant \in [0, \pi]$$

$Cl3 \coloneqq Cl1 == 0$

$Cl4_{3,6,9,10} \coloneqq Cl2_{3,6,8,9,10}\%\pi == 0$

$Cl4_{5,8} \coloneqq Cl2_{5,8} == \frac{\pi}{2}$

$Cl5_{3,6,8,9,10} \coloneqq Cl2_{3,6,8,9,10} == \frac{\pi}{2}$

$Cl5_5 \coloneqq Cl2_5\%\pi == 0$

$Cl6 \coloneqq Cl1 == 0$

$Cl7 \coloneqq Cl1 == 0$

$Ch1 \coloneqq Cl4 + Cl6$

$Ch2 \coloneqq Cl4 + (Cl1 == 0)$

$Ch3 \coloneqq Ch1 + Cl1$

$Ch4 \coloneqq Ch2$

$Ch5_{22,25,26} \coloneqq (Cl2 == constant1) + (Cl1 == constant2)$

Here means the pattern of key entities around a circle

$Ch5_{23} \coloneqq (Cl1 == constant) + Ch1$

Here means the pattern of key entities along a line.

We can see from all the constraints presented above that there are two most basic constraints on which all other constraints are built: Distance (Cl1) and Angle (Cl2). We cannot build any numbers of constraints between any two components, because as we increase the number of constraints there will be a moment that one component is over constraint. So in our approach we decided to consider it as a constraint satisfaction problem. How to solve this problem will be presented in the following section.

CONSTRAINT SATISFACTION PROBLEM (CSP)

As presented in [Bra99], A CSP consists of:

- A set of variables $X = \{x_1, \ldots, x_n\}$;
- For each variable x_i, a finite set D_i of possible values (its domain);
- A set of constraints restricting the values that the variables can simultaneously take.

In our case, the CSP could be defined as below:

$\boldsymbol{CSP} \coloneqq (X, D, C), where$

$X = \{rf_i\}/i \in [0, n], i, n \in \mathbf{N}, where$

$\quad rf_i = (P_i, R_i, S_i)$

$D = \{d_j\}_{j\in[0,n]}, j \in \mathbf{N}, where$

$\quad d_j = (R^3, [0, \pi]^3, R^3)$

$C = \{c_k\}_{k\in[0,m]}, k, m \in \mathbf{N}, where$

$c_k \in Constraints$

Different algorithms could be applied to solve the problem as presented in [Rob]. In general, there are three types of solution:

- Just one solution, with no preference as to which one;
- All solutions;
- An optimal, or at least a good, solution, given some objective function defined in terms of some or all of the variables.

In this paper, we are using the third type of solution, in this case, we proposed a set of objective functions.

$F = (Fp, Fr, Fs), where$

$$Fp = \sum_{i=0}^{i=n} \mu p_i \cdot \left\|P_i^f - P_i^0\right\|^2, \mu p_i \in \mathbf{R}, i \in \mathbf{N}$$

$$Fr = \sum_{i=0}^{i=n} \mu r_i \cdot \left\|R_i^f - R_i^0\right\|^2, \mu r_i \in \mathbf{R}, i \in \mathbf{N}$$

$$Fs = \sum_{i=0}^{i=n} \mu s_i \cdot {S_i}^2, \mu r_i \in \mathbf{R}, i \in \mathbf{N}$$

$$S_i = \begin{cases} \left\|\frac{S_i^f}{S_i^0}\right\|, if \ \left\|\frac{S_i^f}{S_i^0}\right\| > 1 \\ \left\|\frac{S_i^0}{S_i^f}\right\|, if \ \left\|\frac{S_i^f}{S_i^0}\right\| \leq 1 \end{cases}$$

Fp is the positioning energy of all the component from the initial position P_i^0 to the final position P_i^f.

Fr is the rotation energy of all the component from the initial rotation R_i^0 to the final rotation R_i^f.

Fs is the scaling energy of all the component from the initial S_i^0 scale to the final scale S_i^f.

Our objective is to minimize these three energies. And $\mu p_i, \mu r_i \ and \ \mu s_i$ are the three energy factors for each component. If we don't want one specified component change too much its position we can set μp a very large value. In this sense, we build a link of the transformation of reference frame of each component with a semantic meaning. In another word, our CSP is more meaningful compared with typical CAD CSP. The importance of semantics for the constraints could be found in [Tut08]. We can use different factors for different components, we could also use a global factors for all component. The factors actually limit the flexibilities of positioning, rotating and scaling of each component.

With the objective function, the solution of our CSP becomes a numerical optimization problem. We are using a numerical software called Mathmatica to solve the optimization problem, the result could be found in the next section.

RESULTS

The approach presented in this paper has been implemented in a 3D application called Unity3D, which is a famous tool to develop 3D environment. The inputs of our system are different types of multimodal data which has already the geometry (mesh for 3D data, contour for 2D data) and structure information associated. The aim of the test is to evaluate our approach and to see how the three energy factors will affect the final result of the CSP.

Objective: Create a table from multimodal data

Components:

We considered three components to be tested: a leg of a dog from a 3D data, the upper part of a pillar from a 2D data and a cube generated in Unity3D.

Component 1: 2D image with structure– upper part of a pillar. (Fig. 4)

Component 2: 3D mesh with structure - One leg of a dog (Fig.5)

Component 3: 3D mesh without structure- Cube (Fig. 6)

Key entities:

e1 $\in E_{pp5}$, e2 $\in E_{pl2}$, (See Fig. 5)

e3 $\in E_{pp5}$, e4 $\in E_{pl2}$, (See Fig. 6)

e5 $\in E_{pp5}$, e6 $\in E_{pp8}$, (See Fig. 5 and 6)

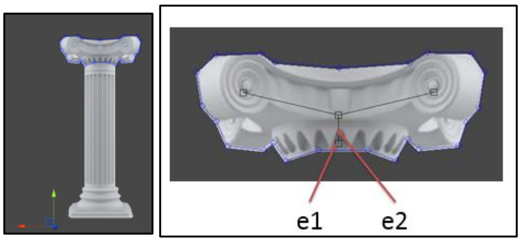

Fig. 4 Key Entity e1 (Epp5) and e2 (Epl2)

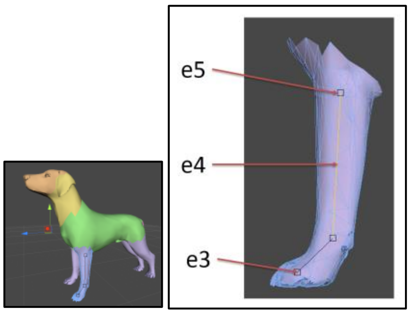

Fig. 5 Key Entity e3 (Epp5), e4 (Epl2) and e5 (Epp5)

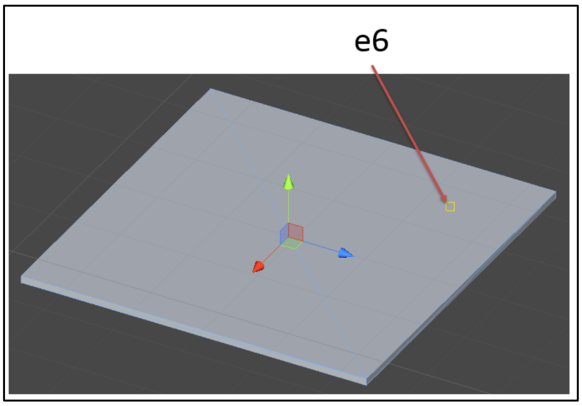

Fig. 6 Key Entity e6 (Epp8)

Constraints:

$c1 = Cl3(e1, e5); c2 = Cl4(e2, e4);$

$c3 = Cl3(e3, e6); c4 = Cl5(e6, e4);$

Results:

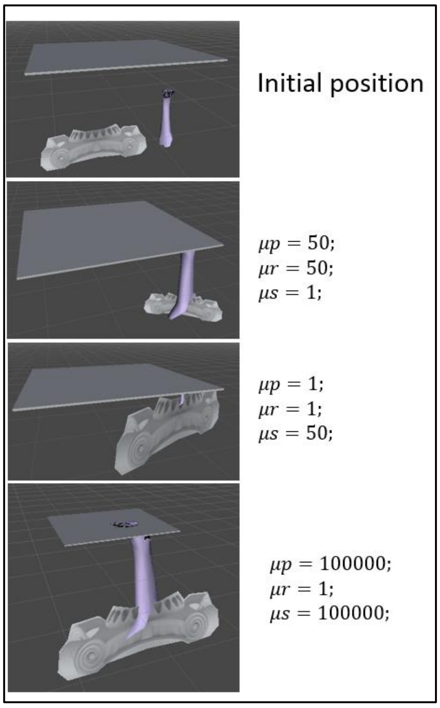

Fig. 7 Result

With the same constraints, we have tested three different configurations of the three energy factors. (Fig. 7) We can see that different configurations of those three factor will allow the components to be rescaled in different ways to satisfy all the constraints.

CONCLUSION

In this paper, we presented our constraint setting and solver engine to be included in our under development system for the specification of objects and scenes from multimodal data using either a traditional keyboard and mouse or a multi-touch UI In the final application the specification of most of the key elements will be derived by the user actions. To facilitate the interaction constraints have been named using semantically meaningful terms. They have been distinguished in low level and high level constraints. Low level constraints are expressed by a single equation, and are then combined to build the high level ones.

Future work will also focus on the integration of deformation possibilities during the constraint satisfaction and on the specification of more meaningful and complex relation between components such as group, father-child and use the constraints to describe those relations.

Acknowledgment

The work has been partially supported by the VISIONAIR project funded by the European Commission under grant agreement 262044, the Co-DIVE project and the Progetto Premiale Tecnologie e sistemi innovativi per la fabbrica del futuro e Made in Italy.

REFERENCES

[Hsu97] Hsu C., Alt G., Huang Z., Beier E., and Brüderlin E., A constraint-based manipulator toolset for editing 3D objects. In Proceedings of the fourth ACM symposium on Solid modeling and applications (SMA '97). New York, pp.168-180.1997

[And95] Anderl R. and Mendgen R., Parametric design and its impact on solid modeling applications. In Proceedings of the Third Symposium on Solid Modeling and Applications, pp.1-12, Salt Lake City, 1995

[Mar08] Martin A. Smith and Xiaogang Chen, CAD and constraint-based geometric modelling algorithms for 2D and 3D woven textile structures, Journal of Information and Computing Science, Vol. 3, No. 3, pp.199-214, England,2008

[Kra91] Glenn A. Kramert, Using degrees of freedom analysis to solve geometric constraint systems, In Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications (SMA '91), Jaroslaw Rossignac and Joshua Turner (Eds.), pp. 371-378, New York, 1991

[Kir13] Ulrich Kirchmaier, Simon Hawe and Klaus Diepold, A Swarm Intelligence inspired algorithm for contour detection in images, Applied Soft Computing, vol. 13, issue 6, pp. 3118–3129, 2013

[Mor12] Marco Mora, Fernando Córdova-Lepe and Rodrigo Del-Valle, A non-Newtonian gradient for contour detection in images with multiplicative noise, Pattern Recognition Letters, vol 33, issue 10, pp. 1245–1256, 2012

[Pap11] Giuseppe Papari and Nicolai Petkov, Edge and line oriented contour detection: State of the art, Image and Vision Computing, vol. 29, issues 2–3, pp. 79–103, 2011

[PGY99] P.C. Yuen, G.C. Feng and J.P. Zhou, A contour detection method: Initialization and contour model, Pattern Recognition Letters, vol. 20, issue 2, pp. 141–148, 1999

[Bia08] S. Biasotti, D. Giorgi, M. Spagnuolo and B. Falcidieno, Reeb graphs for shape analysis and applications, Theoretical Computer Science, vol. 392, issues 1–3, pp. 5–22, 2008

[Har09] Harish Doraiswamy and Vijay Natarajan, Efficient algorithms for computing Reeb graphs, Computational Geometry, vol. 42, issues 6–7, pp. 606–616, 2009

[Lak07] Lakshman Prasad, Rectification of the chordal axis transform skeleton and criteria for shape decomposition, Image and Vision Computing, vol. 25, issue 10, pp. 1557–1571, 2007

[Tao07] Tao Ju, Matthew L. Baker and Wah Chiu, Computing a family of skeletons of volumetric models for shape description, Computer-Aided Design, vol. 39, issue 5, pp. 352–360, 2007

[Abo91] Abolfazl Sirjani, On representation of a shape's skeleton, Pattern Recognition Letters, vol. 12, issue 3, pp. 149–154, 1991

[Ber09] Stefano Berretti, Alberto Del Bimbo and Pietro Pala, 3D Mesh decomposition using Reeb graphs, Image and Vision Computing, vol. 27, issue 10, pp. 1540–1554, 2009

[LGD10] L. G. Domakhina, Skeleton-based segmentation and decomposition of raster pairs of shapes, Pattern Recognition and Image Analysis, vol. 20, issue 3, pp. 293-302, 2010

[Bra99] Sally C. Brailsford, Chris N. Potts and Barbara M. Smith, Constraint satisfaction problems: Algorithms and applications, European Journal of Operational Research, vol. pp. 557-581, 1999

[Tut08] Tutenel, T., Bidarra, R., Smelik, R. M., and de Kraker, K. J, The role of semantics in games and simulations, ACM Comput. Entertain. 6, 4, Article 57 35 pages, 2008